

AD-750 772

INTERFACE MESSAGE PROCESSORS FOR THE
ADVANCED RESEARCH PROJECTS AGENCY (ARPA)
NETWORK

Frank E. Heart

Bolt Beranek and Newman, Incorporated

Prepared for:

Advanced Research Projects Agency

October 1972

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

**Best
Available
Copy**

Report No. 2468

October 1972

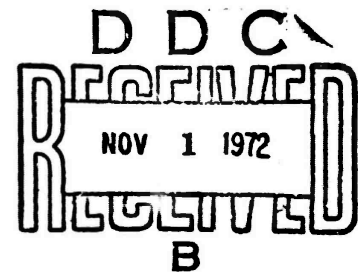
INTERFACE MESSAGE PROCESSORS FOR
THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 15
1 July 1972 to 30 September 1972

Principal Investigator: Mr. Frank E. Heart
Telephone (617) 491-1850, Ext. 470

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 1260

Contract No. DAHC-15-67-C-0179
Effective Date: 2 January 1969
Expiration Date: 31 December 1972
Contract Amount: \$7,517,008



Title of Work: IMP

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. Department of Commerce
Springfield, MA 01115

Submitted to:

Director
Advanced Research Projects Agency
Arlington, Virginia 22209

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Mass. 02138		UNCLASSIFIED	
3. REPORT TITLE		2b. GROUP	
QUARTERLY TECHNICAL REPORT NO. 15			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name)			
Bolt Beranek and Newman Inc.			
6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS	
October 1972	15		
8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)		
DAHC-15-69-C-0179	BBN Report No. 2468		
b. PROJECT NO.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)		
1260			
c.			
d.			
10. DISTRIBUTION STATEMENT			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY	
		Advanced Research Projects Agency Arlington, Virginia 22209	
13. ABSTRACT			
<p>The basic function of the ARPA computer network is to allow large existing computers (Hosts), with different system configurations, to communicate with each other. Each Host is connected to an Interface Message Processor (IMP), which transmits messages from its Host(s) to other Hosts and accepts messages for its Host(s) from other Hosts. There is frequently no direct communication circuit between two Hosts that wish to communicate; in these cases intermediate IMPs act as message switchers. The message switching is performed as a store and forward operation. The IMPs regularly exchange information which: allows each IMP to adapt its message routing to the conditions of its local section of the network; reports network performance and malfunctions to a Network Control Center; permits message tracing so that network operation can be studied comprehensively; allows network reconfiguration without reprogramming each IMP. The Terminal IMP (TIP), which consists of an IMP and a Multi-Line Controller (MLC), extends the network concepts by permitting the direct attachment (without an intervening Host) of up to 64 dissimilar terminal devices to the network. The Terminal IMP program provides many aspects of the Host protocols in order to allow effective communication between a terminal user and a Host process.</p>			

UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computers and Communication						
Store and Forward Communication						
ARPA Computer Network						
Interface Message Processor						
IMP						
Terminal IMP						
TIP						
Honeywell DDP-516						
Honeywell DDP-316						
Multi-Line Controller						
MLC						
Network Control Center						
Host Protocol						
High Speed Modular IMP						
HSMIMP						
Lockheed SUE						
ALOHA						
Satellite Communications						

DD FORM 1 NOV 65 1473 (BACK)

ib

UNCLASSIFIED
Security Classification

2-3147

Report No. 2468

Bolt Beranek and Newman Inc.

INTERFACE MESSAGE PROCESSORS FOR
THE ARPA COMPUTER NETWORK

QUARTERLY TECHNICAL REPORT NO. 15
1 July 1972 to 30 September 1972

Submitted to:

Advanced Research Projects Agency
Arlington, Virginia 22209
Attn: Dr. L.G. Roberts

This research was supported by the Advanced Research Projects
Agency of the Department of Defense under Contract No. DAHC-15-
69-C-0179.

TABLE OF CONTENTS

	Page No.
1. OVERVIEW	1
2. REVISION OF THE MAGNETIC TAPE OPTION	5
3. SATELLITE TECHNIQUES	8
3.1 TDMA	8
3.2 ALOHA	9
3.3 Reservation Systems	10
3.4 BBN ALOHA	11
4. HSMIMP PROGRAM ORGANIZATION	14

1. OVERVIEW

This Quarterly Technical Report, Number 15, describes the aspects of our work on the ARPA computer network during the third quarter of 1972.

During this quarter, three new IMPs were installed and one was relocated. A 316 IMP was delivered to Aberdeen Proving Grounds, Aberdeen, Maryland, and a TIP installed at Computer Corporation of America (CCA), Cambridge. The TIP destined for use at the ICCC show was temporarily installed in another section of the BBN building to allow rehearsal of the scenarios to be presented at the conference. The 316 IMP at McClellan Air Force Base was moved to Xerox Palo Alto Research Center (PARC) when McClellan ceased network activities.

During the third quarter, we continued the efforts of last quarter to bring up the new IMP/TIP software. An operational system was released early in the quarter, and we have been involved in promulgating a series of revisions to eliminate the remaining bugs and add new features. Tuesday mornings, 7:00-8:00 a.m. Eastern time, have been publicly declared as the time set aside to release new versions. This will hopefully allow us to make changes in the system with a minimum of interference to network users.

The new system also incorporated some changes to the IMP-Host protocol. While maintaining, for the most part, backwards compatibility, existing message types were subdivided to allow more precise specifications of the causes of errors and incomplete transmissions. The IMP Going Down message also now specifies the reasons for same and how soon and for how long

the IMP will be down. A new message indicating an interface reset was added. Two obsolete message types were deleted.

We have also engaged in a series of experiments to test and measure the efficiency of the new algorithms. These experiments have not been completed, but have already raised some very interesting questions about the system's performance and, in at least one case, have led to the extermination of a bug.

We have developed some new tools for preparing and documenting large interrupt-driven systems, such as the IMP and TIP. The listings output by these processes, by explicitly classifying each instruction or data word as to physical and virtual interrupt level, aid in debugging and have the potential of allowing some algorithmic (perhaps automated) program verification.

A significant portion of the revisions to the new software has concerned the TIP software. A new protocol for the magnetic tape option was specified and implemented. This and other work on this option are described in Section 2. Major efforts were made to increase communication to and from the users of the TIP. A revision of the IMP/TIP Operating Manual (BBN Report No. 1877), two revisions of the TIP User's Guide (BBN Report No. 2183), and a new document, "Specifications for the Interconnection of Terminals and the Terminal IMP" (BBN Report No. 2277), were issued. Two letters detailing new features and proposals for more of the same were distributed to the network community. A new command, NEWS, allows users dynamic access to current status via facilities of BBN Hosts. Several other new commands were implemented and the logger was restructured. The internal structure of the TIP program was extensively revised to clean up the interrupt structure and reduce interdependence with the IMP program.

During this quarter, we have continued our investigations into the methods of implementing communications links via satellite. We participated in a meeting on satellite techniques at UCLA and have contributed several papers to the ARPANET Satellite System (ASS) series. We have investigated through analysis and simulations several alternative schemes for using a satellite channel for broadcast communication. Some of these proposals are discussed in Section 3.

Another major continuing effort is the development of the HSMIMP. With the selection of the processor and the gross architecture firmly established (see Section 3 of our Quarterly Technical Report No. 14), more detailed work has proceeded. Four members of our hardware group spent considerable time at Lockheed Electronics Company learning the detailed characteristics of the SUE system and fabricating prototype modules. We have received the initial shipment of Lockheed modules for our prototype systems, which are now being used for hardware and software debugging. A cross-assembler for the SUE machine language that runs on our PDP-1 was also developed. Specifications for many of the individual BBN-designed modules have been established; design and (in some cases) construction are in progress. At the same time, we have been very concerned with broader system issues, not only those dealing with making an efficient and reliable multiprocessor, but also those having to do with the IMP/TIP system and subnet as a whole. Our conception of the HSMIMP program organization is discussed in Section 4.

We have also continued our support of the proposed DCA Network. A new member of our group has been specifically assigned to serve as the interface to DCA and has been assisting with the specification of the Host Interface which will be required (for the Honeywell H-6000 computer) as well as with other implementation details.

Finally, we have been concerned with preparations for the ICCC show in October. As mentioned above, the TIP which will be on display was made available for checkout and practice with the various terminals which will be used. We have also participated in several planning meetings and will be sending a support staff to the show.

2. REVISION OF THE TIP MAGNETIC TAPE OPTION

During the past quarter, it was decided that the initial implementation of the TIP magnetic tape option required significant revision before it would be reliable. Thus, after consultation with the ARPA office and the two sites having the option, the magnetic tape option was temporarily decommitted, revised, and rereleased.

The major areas of revision were 1) to the TIP/magnetic tape option interface, 2) to the magnetic tape drive handling routines, and 3) to the tape transfer protocol.

- 1) The TIP/magnetic tape option interface was changed so that many more of the standard TIP mechanisms could be used -- this required two logical TIP ports (62 and 63) to be dedicated to the magnetic tape option. However, this change simplified the interface and made it more reliable since more of it is standardly used on all TIP functions.
- 2) The magnetic tape drive handling routines were simplified by simplifying the record buffering algorithms.
- 3) The tape transfer protocol was completely redone as discussed in the succeeding paragraphs.

By the time it was decided to make this revision of the magnetic tape option, the proposed file transfer protocol originally adopted by the TIP tape option had been discarded by the Network Working Group. Rather than wait for another file transfer protocol to be decided upon, we invented what we have called the TIP magnetic tape transfer protocol. This protocol has the disadvantage of being non-standard, but has the very real advantage

of being almost trivial to implement both on the TIPs and on any Host wanting to communicate with a TIP mag tape.

The magnetic tape protocol is as follows:

- It is built upon the Host/Host protocol.
- A record is an integral number of 16-bit words long.
- The TIP reading a magnetic tape presently sends no more than one record per message, although a record may be more than one message long.
- The TIP writing a magnetic tape can receive multiple record messages. Messages of records received by the TIP from the net are treated as a stream of 8-bit bytes and message boundaries can occur on any byte allowed by Host/Host protocol.
- The message format requires each data record to be preceded by a one-byte code (260_8) indicating that the ensuing data stream (which may be spread over several records) is a magnetic tape record. The code is followed by a two-byte count of the number of 16-bit words in the record. Of course, each message begins with the usual Host/Host protocol (which is not included in the count). Each record is terminated by a zero byte.
- An end of file indication is sent as a record of length zero, i.e., the mag tape code byte (260_8), a count of zero, and the zero byte terminator.
- A 16-bit data word in a magnetic tape transfer message currently contains two 6-bit frames of tape, each packed in the low-order six bits of a byte. Eventually, these 16-bit words may be packed full for greater efficiency.

Both as part of the testing process and also to help make the magnetic tape option immediately more useful, we have written a simple program to run on a TENEX system to copy data between TENEX files and a magnetic tape on a TIP. This program is available for general use, but we do not intend to support it.

3. SATELLITE TECHNIQUES

In preparation for the incorporation of satellites into the ARPA Network, we have been considering techniques for using a satellite channel in a way which:

- 1) is compatible with the packet switching technology of the ARPA Network,
- 2) is highly reliable, and
- 3) allows high utilization of the channel without subjecting messages to too large a delay.

It seems clear that the satellite channel should be a broadcast channel because this is inherently in the spirit of packet switching. A broadcast channel would allow the commingling of packets for different destinations on the same channel, therefore producing a more efficient utilization of the channel. The techniques for governing this mixture of packets include Time Division Multiple Access (TDMA), ALOHA, various reservation systems, and a new system developed at BBN. In the following sections we will discuss each of these techniques.

3.1 TDMA

TDMA is a well-known mechanism for allowing many nodes access to one channel. The scheme is to divide time into cycles, with the cycles subdivided into slots. Each node is allocated the same slot in each cycle. Its basic disadvantage is that the fraction of the channel which is available to one node is inversely proportional to the number of nodes using the channel, rather than proportional to the fraction of the channel which is not being used.

3.2 ALOHA

The ALOHA system was introduced by the University of Hawaii. It operates in this way:

1. When a packet is ready for transmission, transmit it.
2. If you do not receive that packet at the appropriate time, wait for a random length of time (in order to avoid recollision) then retransmit it. Then go to 2.

The ALOHA system is a simple method of dynamically allocating the channel without centralized control. This dynamic allocation can be a significant improvement over the fixed allocation of TDMA, when the nodes have different demands on the channel.

The major disadvantage is that the ALOHA system, as described, has a theoretical capacity of only .18, and a corresponding delay of 2.7 times the round trip time to the satellite at capacity. In order to improve these figures, we may add the constraint of starting the transmission of packets only at discrete time intervals (the beginning of a slot). The addition of this constraint doubles the capacity of the channel.

The average delay due to collisions and consequent retransmission experienced by a packet using the satellite channel with slots is a function of the traffic on the channel. When the throughput is one tenth to one fifth of the channel, the average delay is very nearly one round trip time. However, as the throughput approaches .36, the delay rises rapidly to 2.7 times the round trip time. If the ground stations try for a throughput greater than .36, the delay increases greatly, and the throughput decreases. This scheme would be adequate if we could afford to use only a small portion of the channel.

3.3 Reservation Systems

Reservation systems are those in which some slots in the channel are scheduled on the basis of requests by the nodes. These requests are called bids, and the result (if successful) is a reservation. In the systems of interest to us, the bids are made by using ALOHA in the portion of the channel which is not being used for reservations, or by using the current reservation.

The basis for reservation systems is that if the exact time-varying load on each of the nodes is known, it should be possible to schedule the channel for optimum utilization and delay. In theory, this scheduling can produce a channel capacity of unity. There are, however, several problems.

- Any system which requires tight scheduling of the channel is sensitive to perturbations in the form of line errors or new nodes. If a bid is correctly received by all nodes but one, that node will have an incorrect idea of how to schedule the channel. This problem may be reduced by using error correction, etc., but a mechanism will have to exist to reset everything if things get too bad. Furthermore, since a new node would have no information, it would probably have to force a reset.
- There are two basic reservation strategies, predictive and non-predictive. In a non-predictive system, one waits until a packet is ready for transmission before making a reservation for it. This allows higher throughput; but the minimum delay is twice the round-trip time. We have done simulations on this.

In a predictive system, one tries to anticipate the amount of traffic that will be ready for transmission at the time that node's bid is honored. The difficulties here are:

- Each node must keep arrival statistics and determine its prediction. This costs machine bandwidth.
- When, and to the extent that, the predictions are wrong, either reserved space will go unused, or surplus packets will have to be queued. This problem becomes more pronounced if the source distribution has a large variance.

We expect that in the ARPA Network, the routing methods employed will tend to smooth the traffic.

- Also, a node is forced to reserve more than it expects to receive to avoid the problem that occurs when a queue has the same service rate as arrival rate: the length of the queue is indeterminate and all traffic would be delayed. This trouble was experienced in the ARPA Network with an earlier routing algorithm.

We expect that the sensitivity of reservation systems to perturbations is the greatest difficulty with these schemes; it should not be taken too lightly.

3.4 BBN ALOHA

This idea is a variation on Reservation schemes, but avoids their major difficulty: keeping reservation information accurate. In addition, it allows the steady-state portion of a node's traffic to pass with a capacity of unity, and near minimum latency. It allows the variable portion of the traffic to trade delay for bandwidth.

This scheme avoids the basic limiting phenomenon of the ALOHA system (collisions) in a straightforward manner. In order to decrease the probability of collision for a packet, each node uses the slots which it successfully used last time (whatever "last time" means). The probability of collision will be significantly lower for the packets which use those slots.

In order to clarify the notion of "last time", here is an example of how to implement this system.

Let T be an arbitrary fixed number of slots. Let each node keep a history of slot usage for the past T slots; the slots which are en route over the satellite link are included in T , but are not included in the history until they are received.

When a node has a packet to transmit, it should try to wait for a slot which its history indicates was successfully used by that node T slots ago. If this would cause too large a delay or too small a throughput, then wait for a slot which was empty T slots ago.

The correct value of T probably depends on time constants in the network, because T affects the response time of this channel. As a guideline, T should probably be much greater than the number of nodes. Perhaps T should be an integral multiple of the routing period so that routing messages always use the same slots, and have low delay.

We have done analysis and simulation of this scheme which verify that a capacity of unity can be obtained for the portion of the source rate which is constant.

There were two difficulties with this scheme:

- It is easier to lose a slot than to gain one. This decreases the extent to which the variable portion of the traffic is smoothed, and decreases the advantage of this scheme when used with sources which have a large variance.
- There is no implicit mechanism for ensuring fairness in utilization of the channel. The big guys can squeeze the little guys out.

The first of these difficulties has been attacked by sending some artificial traffic into a node's slots when they would otherwise be lost due to a temporary lull in the source traffic. This artificial traffic has been simulated with good results, using a level which makes it as difficult to lose a slot as it is to gain it back.

Fairness, as we have seen in the rest of the network, has been a difficult concept. It is probably best approached via an explicit algorithm in each node. A possible algorithm would have nodes which occupy a significant portion of the channel avoid decreasing the fraction of empty slots beyond a certain level, thus providing a way for nodes with lighter traffic levels to enter the channel.

Conclusions

We have reached the conclusion that TDMA and Reservation schemes are probably ill-suited for use in this environment. We believe that BBN ALOHA is the best of these schemes for IMP-IMP communication via satellite.

4. HSMIMP PROGRAM UTILIZATION

Classical computers rely on an interrupt structure to allow synchronization of real-time events with a minimum of overhead. The important ingredients of an interrupt system, insofar as we are concerned, are a path by which the processor can be notified to divert its attention and a mechanism by which the context (environment) of the processor can be remembered. In a multi-processor system, the former is complicated not only by the physical distribution of the processor but also by the problem of deciding which processor should handle the interrupt. We have also observed that in a program as heavily interrupt driven as the IMP, 14-15% of the machine bandwidth does only context saving and restoration. This is particularly painful in the IMP system where one can find very many places in which there is no context necessary to the execution of the next step (except perhaps the program counter).

To overcome these disadvantages, we have developed a different sort of program organization which obviates the need for classical interrupts. The program is divided into "strips", pieces of code which take less than 200-300 μ sec to execute and which require no processor context upon entry. The stimulus for executing a strip is usually either a real-time event (e.g., I/O completion) or a software event in the middle of another strip. Each strip is assigned a number.

The dispatch of processors to strips is handled with the aid of the Pseudo-Interrupt Device (PID), a BBN-designed module which probably resides on an I/O bus. When a device interface needs service, it writes the number of the strip which performs its service routine to the PID. Similarly, when a processor

finishes a strip which is logically continued in another strip, the processor writes the number of the latter to the PID. When a processor needs a new task, it does a read from the PID. The PID returns the highest number written to it but not yet read, i.e., it tells the processor what the highest priority task waiting for service is and removes that task from the list of those waiting. The processor can then dispatch to the appropriate routine with a minimum of overhead (2-3%).

Thus we have avoided the difficulties of a classical interrupt system while, at the same time, solving the problem of task assignment to processors. By imposing a maximum on strip times, we can guarantee that the latency before the service routine can be started for the highest priority task is no greater than this maximum time, and we expect the average time to be the average strip length divided by the number of processors. This must be contrasted with the context saving time of the classical system. On the other hand, we are spared the cost of a more traditional polling scheme since the PID serves as a central clearing house; a processor need only poll in one place.

This program structure is very nicely suited to a multi-processor. Each processor can and generally does execute all tasks, thus making the scheme extremely modular. Task assignment is handled in a simple, generalized, inexpensive manner. *Without any changes to the program*, a machine may be configured with any number of processors from one to some reasonable number. If a processor fails (or is shut down for maintenance) in a multi-processor HSMIMP, the machine continues, albeit at a smaller bandwidth, without requiring any dynamic reconfiguration or the like.